سلسلة كن أسدا للإبداع

BE LION CREATIVITY

مدخل إلى الإكس. أم. ال

(XML , DTD , CSS , XSL)

خالد السعداني



مدخل إلى الإكس -أم-أل

(XML , DTD ,CSS, XSL)

من إعداد : خالد السعداني



كتب للمؤلف:

سبيلك المختصر لتعلم السي شارب **Csharp**

مدخل إلى الداتا أكسيس لاير

تعلم كيف تصنع **NotePad** متطور عن طريق الفيجوال بسيك

تحزيم برامج عامل/خادم في الفيجوال ستوديو (متوفر بالفرنسية)

مدخل إلى الإكس – أم – أل



لمحة عن الكاتب

الإسم الكامل : خالد السعداني

الهاتف :0673-07-51-05

من مواليد 18/05/1989 بمدينة الفقيه بن صالح ، المغرب

تقني متخصص في التنمية المعلوماتية

الكفاءات المهنية : البرمجة ب :

VB.NET, C#.NET, C++, C, JAVA(J2EE)

وبالنسبة للويب :

ASP.NET, PHP, HTML, JAVASCRIPT, AJAX, CSS, JQUERY, XML

أما قواعد البيانات :

MS ACCESS, MS SQL SERVER, MYSQL



بسم الله الرحمان الرحيم

أحبابي الأفاضل،

نظرا لقلة المراجع والدروس العربية التي تتحدث عن XML و توابعه، ارتأيت أن أخرج عن سياق السلسلة التربوية والمعرفية "كن أسدا" بغرض سد هذه الثغرة ولو بمعلومات ضئيلة في انتظار تأليف مرجع شامل، على العموم فهذا العمل المتواضع كاف لسد حاجيات المتطلعين إلى معرفة XML و DTD و XSL .

وحتى يكون النفع كبيرا أنصح بأن يكون القارئ على اطلاع بأساسيات HTML حتى لا يلقى أي صعوبات في الفهم.

ولا ننسى كما جرت العادة أن ننبهكم إلى أن "لكل شيء إذا ما تم نقصان" ، فأدنى خطأ تتمكنون من العثور عليه بين طيات الكتاب لا تتوانوا في إرشادنا إليه ولكم منا جزيل الشكر .

أتوقف عند هذا الحد حتى لا أكون ضيفا ثقيلا، و أتمنى لك –أيها القارئ– مسيرة موفقة في حياتك، واللهم انصر الإسلام و المسلمين !

خالد السعداني



"يا أيها الذين آمنوا اتقوا الله و قولوا قولا سديدا. يصلح لكم أعمالكم و يغفر لكم ذنوبكم ومن يطع الله و رسوله فقد فاز فوزا عظيما"

الأحزاب : 70 و 71



للمراسلة من اجل طرح الاقتراحات والملاحظات

والتساؤلات نقدم لكم عنواننا الالكتروني :

Khalid_ESSAADANI@Hotmail.fr

بسم الله على بركة الله



مصطلح XML هو اختصار ل Extensible Marquap Language، إذا ما قمنا بترجمة حرفية لهذه العبارة فسنحصل على "لغة الوسوم الإمتدادية" .... ☹ وماذا يعني ذلك؟؟؟؟؟

حسنا، إذا كنت على اطلاع ببرمجة المواقع وسبق لك أن احتككت بشيء اسمه HTML فأنا على يقين بأنك تعرف ماهو الوسم (بالفرنسية Balise وبالإنجليزية Tag)، نفس الأمر في xml فهي لغة مبنية على الوسوم..

مثال لوسم في HTML:

```
<center>  Ce texte sera affiché dans le centre </center>
```

كما تلاحظ فإن الوسم له بداية ونهاية وبين البداية و النهاية نضع العبارة التي نريد تطبيق الوسم عليها.

☹ ربما فهمنا معنى الوسوم لكن ماذا تقصد ب "الإمتدادية" ؟؟؟؟

تختلف الوسوم في xml عن الوسوم في HTML في كون هذه الأخيرة ثابتة ولاتسمح لنا بإنشاء وسوم حسب رغبتنا، أما XML فتمنح للمستعمل القدرة على إنشاء وسوم خاصة به وهذا ما يقصد بالوسوم الإمتدادية.

متطلبات XML:

لإنشاء ملف XML سنحتاج إلى محرر نصوص مثل Bloc Note المرفقة مع Windows وإلى متصفح من أجل مشاهدة النتائج.

الغرض من XML هو حفظ البيانات ونقلها بطرق سلسة، ويتوفر ملف XML على بنية شجرية تنازلية، حيث يبدأ من الجذر ثم يمتد إلى عناصر جزئية.



وحتى نستوعب المثال جيدا، لنفترض أن لدينا قاعدة بيانات Database تسمى Vente تتوفر على جدول اسمه Article ، هذا الأخير الذي يتوفر بدوره على الحقول التالية:

- رقم المنتج
- اسم المنتج

أي أن قاعدة البيانات ستكون هكذا :

قاعدة البيانات vente

Code Article	Libelle Article
1	Clavier
2	Souris



من أجل إنشاء ملف xml مناسب فإن الكود سيكون هكذا:

```
<?xml version="1.0"?>
<vente>
  <articles>
    <article>
      <code>1</code>
      <libelle>clavier</libelle>
    </article>
    <article>
      <code>2</code>
      <libelle>Souris</libelle>
    </article>
  </articles>
</vente>
```

بالنسبة للسطر الأول فغرضه هو إعلام المتصفح بأن ما نحن بصدد عرضه هو عبارة عن xml

ملاحظة:

يجب الأخذ بعين الإعتبار حالة أحرف الوسوم Les balises فهناك فرق بين الحروف الصغيرة والكبيرة.

كما لايصح استعمال بعض الرموز ك & و > وغيرها، ولكن يتم تعويضها بما يلي:

- & → &
- < → <
- > → >
- ' → '
- " → "



أي أن الشفرة التالية خاطئة:

```
<code> 3>2 </code>
```

والصحيح هو:

```
<code> 3&gt;2 </code>
```

وهذا مثال آخر على ملف XML مختلف نسبيا على الملف الأول:

```
<?xml version="1.0"?>
<vente>
  <articles>
    <article code="1">
      <libelle>clavier</libelle>
    </article>
    <article code="2">
      <libelle/>
    </article>
  </articles>
</vente>
```

سنقوم الآن بشرح هذا الملف:

كما سبق أن قلنا فالسطر الأول يعلم المتصفح بأن الشفرة التي نريد تنفيذها عبارة عن XML.

السطر الثاني عبارة عن إعلان للجذر وهو vente وكما ترى به نبدأ وبه نختم.

السطر الثالث يمثل الإعلان عن العنصر Articles الذي بدوره يتوفر على عناصر فرعية.



السطر الرابع <article code="1"> يقدم لنا مفهوما جديدا يطلق عليه اسم الخاصية Attribut ويكون متصلا بالعنصر، كما هوالحال مع HTML بحيث نجد بعض الوسوم تتوفر على خصائص مرتبطة بها كما يعرض هذا المثال:

```
<font color="Red"> Ce texte sera coloré en rouge </font>
```

أما باقي سطور الملف فكتبت بطريقة عادية بحيث نفتح الوسم ونقفله ماعدا السطر الأخضر:

```
<libelle/>
```

فكما تلاحظ هذا الوسم أو هذا العنصر لا يتوفر على وسم ختام ولا على محتوى، غير أن ما يميزه عن باقي الوسوم هو احتواؤه لرمز القسمة / عند نهاية السطر....وذلك هو السر !

بحيث أن هذا الوسم مفتوح ومغلق في آن واحد ويكون ذلك بسبب أنه لا يتوفر على محتوى وهو يعادل السطر التالي:

```
<libelle><libelle/>
```

ومن باب اختزال الشفرة فضلنا الطريقة الأولى.

التعاليق في XML

الغرض من التعاليق أو التعليقات Commentaires/Comments هو السماح للمبرمج بوضع نص غريب عن لغة البرمجة من أجل شرح بعض الشفرات أو من أجل تذكرها.



لكتابة التعاليق يجب مراعاة الصيغة التالية:

```
<!--هذا عبارة عن تعليق-->
```

مفهوم DTD:

DTD هو ملف يحتوي على مجموعة من القواعد والضوابط التي على ملف xml اتباعها، وهو موضوع شاسع لا يمكننا حصره في بضعة أسطر وسنكتفي فقط بأخذ القسط الذي يفيدنا .

من فوائد DTD أنه يمكننا من تنظيم شفرة XML بحيث من خلاله نستطيع تحديد عناصر ملف XML وعددها وترتيبها وماإلى ذلك.

إن كنت على اطلاع بقواعد البيانات فستفهم أهمية DTD بحيث أنه يضمن لملف XML القواعد والشروط التي يجب أن يحترمها، كعدد تكرار بعض العناصر وقيمها الإفتراضية.

هناك نوعان من DTD الأول داخلي Interne والآخر خارجي Externe :

1. DTD داخلي:

ويكون داخل ملف xml وصيغته كالتالي:

```
<!DOCTYPE Racine [
        Liste des éléments.
        ]>
```



بحيث Racine هو الجذر الرئيسي، ثم يأتي بين المعقوفتين سرد العناصر والخصائص.

أي إذا أردنا أن ننشئ DTD مناسب لملف XML الذي سبق وأنشأناه فسيكون كما يلي:

```
1. <?xml version="1.0"?>
2. <!DOCTYPE vente[
3. <!ELEMENT vente (articles)>
4. <!ELEMENT articles ( article+)>
5. <!ELEMENT article (code,libelle)>
6. <!ELEMENT code (#PCDATA)>
7. <!ELEMENT libelle (#PCDATA)>
8. ]>
<vente>
  <articles>
    <article>
      <code>1</code>
      <libelle>clavier</libelle>
    </article>
    <article>
      <code>2</code>
      <libelle>Souris</libelle>
    </article>
  </articles>
</vente>
```

السطر الثاني هو بداية DTD ويكون مرفقا بالجذر.

السطر الثالث يضم العنصر الذي يأتي مباشرة تحت الجذر.

السطر الرابع يضم العنصر الفرعي ووضعنا أمامه علامة زائد + كي نسمح بتكرار العنصر أكثر من مرة، ويمكننا وضع الإشارات التي يعرضها الجدول التالي حسب حاجتنا.



الرمز	الغرض منه
,	تعني (أيضا and) مثلا: (code,libelle)
|	تعني(أو or) مثلا (code | libelle)
?	تعني أن هذا العنصر اختياري مثلا(code , libelle?)
+	تعني واحد أو أكثر مثلا (code,libelle+)
*	تعني أي رقم من صفر فما فوق مثلا (code, libelle*)

أما فيما يخص هذا السطر :

```
<!ELEMENT code (#PCDATA)>
```

فيعني أن العنصر الفرعي code يمكن أن يضم بيانات من نوع حرفي أو رقمي.

2. DTD خارجي:

ويكون بنفس صيغة DTD الداخلي إلا أنه لايضم سطر البداية:

```
<!DOCTYPE vente[
]>
```

نقوم بكتابة نفس الأسطر ونحفظ الملف بامتداد DTD. كما يعرض المثال التالي:

```
<?xml version="1.0"?>
<!ELEMENT vente (articles)>
<!ELEMENT articles ( article+)>
<!ELEMENT article (code,libelle)>
<!ELEMENT code (#PCDATA)>
<!ELEMENT libelle (#PCDATA)
>
```



ثم نناديي عليه من داخل ملف XML بالطريقة التالية:

```
<?xml version="1.0"?>
<!DOCTYPE vente SYSTEM "fichier.dtd">
<vente>
  <articles>
    <article>
      <code>1</code>
      <libelle>clavier</libelle>
    </article>
    <article>
      <code>2</code>
      <libelle>Souris</libelle>
    </article>
  </articles>
</vente>
```

بحيث vente هو اسم الجذر و fichier.dtd هو اسم ملف DTD.

إذا لم يكن ملف DTD متواجدا مع ملف XML في نفس المجلد يتوجب عليك أن تكتب مسار ملف DTD كاملا.

إظهار XML بواسطة CSS:

كما ترى فشكل XML على المتصفح غير جذاب ..... ☹ أردت قولها من الأول، فهو بشع !!!!

لا تترعج فيمكننا التحكم بمظهر وسوم XML كما نريد إما من خلال CSS أو من خلال XSL.

إن كنت تسمع لأول مرة ب CSS (Cascading Style Sheets)، فهي عبارة عن لغة تستخدم غالبا مع HTML من أجل تحسين عرض الصفحة وشكل محتواها.

الآن سننشئ ملف CSS الذي سيسمح لنا بتغيير هيئة ملف XML:

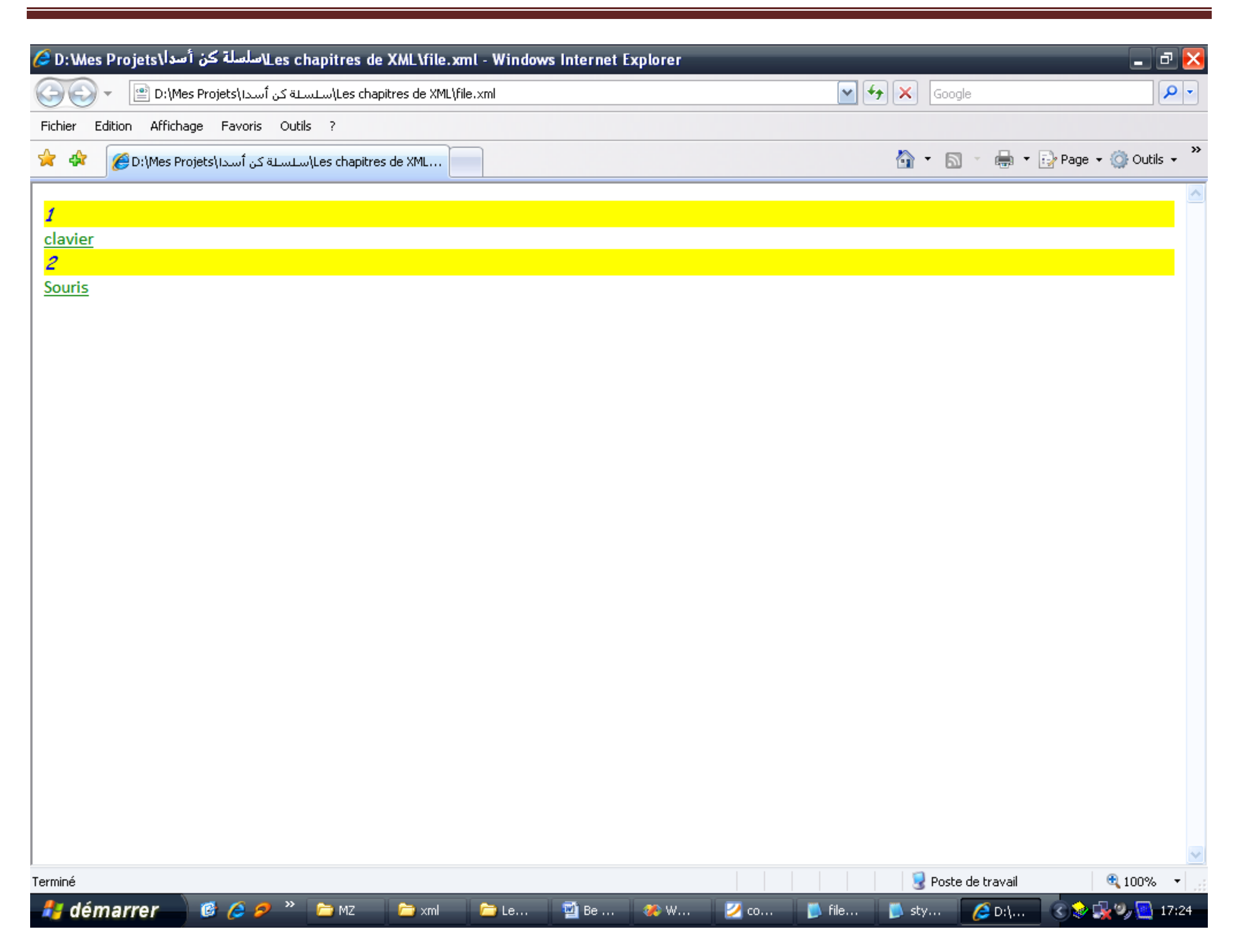



```
code
{
	background-color: Yellow;
	display: block;
	color:Blue;
	font-family:Comic Sans MS;
	font-style:italic;
}
libelle
{
	display: block;
	color:Green;
	font-family:Calibri;
	text-decoration:underline;
}
```

كما ترى وضعنا العناصر التي نرغب في تغيير شكل إظهارها، وهي العنصر code و libelle، بعد ذلك نقوم بحفظ الملف بامتداد CSS. ثم ندرجه داخل ملف XML كما يلي:

```
<?xml version="1.0"?>
<?xml-stylesheet href="css.css" type="text/css"?>
<!DOCTYPE vente SYSTEM "ok.dtd">
<vente>
  <articles>
    <article>
      <code>1</code>
      <libelle>clavier</libelle>
    </article>
    <article>
      <code>2</code>
      <libelle>Souris</libelle>
    </article>
  </articles>
</vente>
```

عند الحفظ والولوج إلى ملف XML فإن النتيجة ستكون كما يلي:



أعتقد أنك أدركت أهمية لغة CSS بعد رؤية النتيجة، إن كان الأمر كذلك فحاول أن تبدع مظهرا جديدا من اختيارك .



إظهار XML بواسطة XSL:

XSL هو نوع من اللغات التي تهتم بمظهر البيانات، تماما ك CSS إلا أنه يتوافق مع XML أحسن من CSS، بحيث يقوم بتحويل XML إلى HTML بطريقة خلابة.

سننشئ ملف XSL أولا وبعدها نقوم بإدراجه داخل ملف XML حتى نتأمل النتيجة.

هذا مثال على ملف XSL:

```
<?"xml version="1.0?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <html>
      <body>
        <h3>Liste des articles</h3>
        <table border="1" width="100%">
          <tr bgcolor="yellow">
            <td>Code</td>
            <td>Libelle</td>
          </tr>
          <xsl:for-each select="vente/articles/article">
            <tr>
              <td>
                <xsl:value-of select="code"/>
              </td>
              <td>
                <xsl:value-of select="libelle"/>
              </td>
            </tr>
          </xsl:for-each>
        </table>
        <br/>
        <br/>
        <br/>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```



والآن إلى شرح هذه الشفرة:

```
<?xml version="1.0"?>
```

لغة XSL مشتقة من XML لهذا لهما نفس المقدمة.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

هذا السطر هو البداية الحقيقية لـ XSL بحيث يخبر المتصفح بأن ما سيتم عرضه عبارة عن XSL، أما الخاصية xmlns:xsl فتسمى مجال الأسماء Namespace.

```
<xsl:template match="/">
```

الخاصية match التي تأخذ القيمة / ترمز إلى جذر الملف xml أي أن المعالجة ستبدأ من الجذر ثم تمتد إلى باقي العناصر الفرعية.

ما تبقى من الكود عبارة عن كود HTML ماعدا الوسم التالي:

```
<xsl:for-each select="vente/articles/article">
```

وهي حلقة تكرارية معناها جلب كل عنصر فرعي متواجد تحت العنصر article .

```
<xsl:for-each select="vente/articles/article">
```

ثم نقوم بإظهار القيم في خلايا الجدول عن طريق الوسم الآتي:




```
<xsl:value-of select="code"/>
```

فقط أريد تنبيهك إلى احترام الترتيب وعدم الخلط بين الوسوم، أو كتابة إحدى العناصر بشكل خاطئ، إن اتبعت نفس الخطوات التي قمنا بها فستحصل على النتيجة التالية:

D:\Mes Projets\سلسلة كن أسدا\Les chapitres de XML\file.xml - Windows Internet Explorer

Liste des articles

Code	Libelle
1	clavier
2	Souris

إن حصلت على خطإ، فالمرجو مراجعة ملفي xml و xsl ، ومراقبة حالة الأحرف وترتيب الوسوم وسلامة الشفرة، إن ظل الخطأ قائما فالمرجو مراسلتي عبر الإيميل عن طريق بعث صورة الخطأ وأي ملاحظات ثانوية.



يمكنك XSL من القيام بمجموعة من العمليات الرائعة، مثل جلب المعلومات بشرط، أو بترتيب معين وغيرها من الإجراءات .

وسنستعرض هنا أهم هاته العمليات:

سنقوم أولا بإنشاء ملف XML جديد وسنطبق عليه هاته العمليات.

```
<?xml version="1.0"?>
<Homme>
  <Personne>
    <Person ID="1">
      <Nom>Khal</Nom>
      <Prenom>Simo</Prenom>
      <Age>20</Age>
    </Person>
    <Person ID="2">
      <Nom>MAKBOUL</Nom>
      <Prenom>Hamid</Prenom>
      <Age>20</Age>
    </Person>
    <Person ID="3">
      <Nom>ESSAADANI</Nom>
      <Prenom>Khalid</Prenom>
      <Age>21</Age>
    </Person>
    <Person ID="4">
      <Nom>Bourzayq</Nom>
      <Prenom>Khalid</Prenom>
      <Age>20</Age>
    </Person>
  </Personne>
</Homme>
```

العملية الأولى: جلب كل الأشخاص بدون شرط:




```
<?"xml version="1.0?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
    <html>
      <body>
        <h3>Tous Les personnes</h3>
        <table border="1" width="100%">
          <tr>
            <td>Nom</td>
            <td>Prenom</td>
            <td>Age</td>
          </tr>
          <xsl:for-each select="Homme/Personne/Person">
            <tr>
              <td>
                <xsl:value-of select="Nom"/>
              </td>
              <td>
                <xsl:value-of select="Prenom"/>
              </td>
              <td>
                <xsl:value-of select="Age"/>
              </td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

ستكون النتيجة كما يلي:

Tous Les personnes

Nom	Prenom	Age
Khal	Simo	20
MAKBOUL	Hamid	20
ESSAADANI	Khalid	21
Bourzayq	Khalid	20



العملية الثانية: جلب كل الأشخاص مرتبين حسب الأسماء:

سنقوم فقط بتعويض سطر for-each بالسطر التالي:

```
<xsl:for-each select="Homme/Personne/Person" order-by="+Nom">
```

الرمز + يعني أن يتم الترتيب تزايدي من a إلى z وللقيام بالعكس أي ترتيب تناقصي نستبدل العلامة + بالعلامة –

النتيجة بعد الترتيب التزايدي ستكون كما يلي:

Tous Les personnes

Nom	Prenom	Age
Bourzayq	Khalid	20
ESSAADANI	Khalid	21
Khal	Simo	20
MAKBOUL	Hamid	20

لاحظ معي ترتيب الأسماء كيف تغير تزايديا.

العملية الثالثة:جلب الأشخاص الذين يحملون إسم Khalid:




```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <html>
      <body>
        <h3>Avec condition</h3>
        <table border="1" width="100%">
          <tr>
            <td>Nom</td>
            <td>Prenom</td>
          </tr>
          <xsl:for-each select="Homme/Personne/Person[Prenom='Khalid']">
            <tr>
              <td>
                <xsl:value-of select="Nom"/>
              </td>
              <td>
                <xsl:value-of select="Prenom"/>
              </td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

ستكون النتيجة كمايلي:

Avec condition

Nom	Prenom
ESSAADANI	Khalid
Bourzayq	Khalid

يمكنك القيام بالعديد من العمليات على بيانات XML عن طريق ملف XSL .

سنكتفي بهذا القدر آملين أن نكون قد وفقنا في توصيل الرسالة.



إذا كان لديك ملاحظات أو تساؤلات أو وجدت أخطاء في الكتاب فلا تتردد في أن تراسلني عبر

Khalid_ESSAADANI@Hotmail.fr